Broadcasting Convolutional Network for Visual Relational Reasoning

Simyung Chang[1,2], John Yang[1], SeongUk Park[1], and Nojun Kwak[1](✉)

[1] Seoul National University, Seoul, South Korea
{yjohn,swpark0703,nojunk}@snu.ac.kr
[2] Samsung Electronics, Suwon, South Korea
timelighter@snu.ac.kr

Abstract. In this paper, we propose the *Broadcasting Convolutional Network* (BCN) that extracts key object features from the global field of an entire input image and recognizes their relationship with local features. BCN is a simple network module that collects effective spatial features, embeds location information and broadcasts them to the entire feature maps. We further introduce the *Multi-Relational Network* (multiRN) that improves the existing Relation Network (RN) by utilizing the BCN module. In pixel-based relation reasoning problems, with the help of BCN, multiRN extends the concept of 'pairwise relations' in conventional RNs to 'multiwise relations' by relating each object with multiple objects at once. This yields in $\mathcal{O}(n)$ complexity for n objects, which is a vast computational gain from RNs that take $\mathcal{O}(n^2)$. Through experiments, multiRN has achieved a state-of-the-art performance on CLEVR dataset, which proves the usability of BCN on relation reasoning problems.

Keywords: Visual relational reasoning · BCN · Broadcast · CLEVR Multi-RN · Visuo-spatial features

1 Introduction

A complete cognizance of a visual scene is achieved by relational reasonings of a set of detected entities in an attempt to discover the underlying structure [18]. Reasoning comparative relationships allows artificial intelligence to infer semantic similarities or transitive orders among objects in scenes with various perspectives and scales [5]. While the core of relational reasoning instrumentally depends on spatial learning [3,28], the relational networks (RNs) [27,29] have fostered the performance vastly on related tasks based on their spatial grid features. However, the number of objects in conventional RNs upsurges as their

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01267-0_46) contains supplementary material, which is available to authorized users.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11219, pp. 780–796, 2018.
https://doi.org/10.1007/978-3-030-01267-0_46

method assumes that each grid represents an object at the corresponding position within the scene regardless of the existence of an object at a grid position. Moreover, the computational cost increases quadratically as RNs are based on pairwise computation of objects' relations for relational reasoning.

This computational burden is inevitable for visual reasoning problems if conventional architectures of convolutional neural networks (CNNs) [21] are used. Although, CNNs have allowed success in many computer vision problems [6,7,9,12,19,24], yet they still suffer from difficulties in generalization over geometric variations of scenes. This is mainly due to the receptive fields that are mapped with convolution filters at fixed areas, which derives CNNs to disregard spatial locations in the process of searching for optimal features. Either bigger size of filters that embrace multiple input entities or repetitive usage of smaller filters in deeper networks are typically used to learn spatial relationships. CNNs, however, still show limited performances for large deformations of inputs as long as the receptive fields of convolution or pooling filters stay local and small-sized [13,14,23,25].

In order to learn relationships among objects, the correlation of objects needs to be defined along with the segregation of non-object features. And, CNNs' structural loss of spatial information also needs to be overcome to handle dynamic variations of object sizes and locations. We are motivated to solve such issues through globally extending receptive fields of object features and efficiently learning correlations among objects in an end-to-end manner. To this end, we propose a modular technique named *Broadcasting Convolutional Network* (BCN) that can be applied in any CNNs to enable learning spatial features with absolute positional information, broadcasting the features and analyzing visual relations among given objects. This technique not only overcomes the limitations of conventional narrow-sighted convolution operations by extending the receptive fields ideally to the global manner, but also allows to define a novel neural network called the *Multi-relational Network* (multiRN) that outperforms on the relational reasoning tasks in terms of both performance and computational efficiency. The proposed multiRN achieves the state-of-the-art performance on the CLEVR dataset which is the representative dataset for relational reasoning.

The paper is organized as follows. In Sects. 2 and 3, the proposed BCN and multiRN are explained in detail. In Sect. 4, the novelty of our work is described by comparing the methods with the related works. Section 5 shows experimental results and finally, Sect. 6 concludes the paper.

2 Broadcasting Convolutional Network

In this section, we first describe the overall architecture of BCN with details of our implementations and their purposes. The proposed BCN is depicted in Fig. 1 which mainly consists of three components: (1) coordinate channel embedding (CCE), (2) encoding visual features with objective spatial information and (3) broadcasting globally max-pooled features through expansions.

The BCN is applied after each pixel of the feature maps acquires proper sizes of receptive fields through basic convolutions. The module makes feature

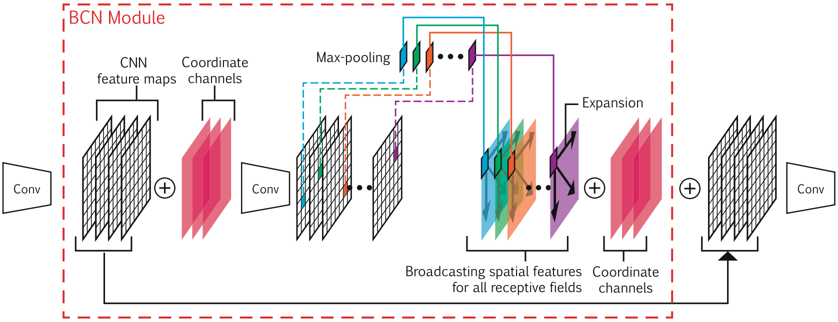

Fig. 1. Broadcasting Convolutional Network Modules. Feature maps acquired from previous convolution layers are concatenated with coordinate planes during CCE phase. Then, additional light-weight convolution layer(s) (*e.g.* two layers of 1×1 convolution with ReLU activation) is(are) applied. An $1 \times 1 \times n$ vector is generated by max-pooling from each filter channel of the n resultant feature maps. The vector is then expanded to emulate the size of the original feature maps, and merged with the original feature maps as an input to the next layer of convolution.

maps that represent the coordinate information and concatenates them with the original CNN feature maps. Then, rather simple convolution operations (*e.g.* a few layers of 1×1 convolution with ReLU activation) are applied, which is followed by a global max-pooling stage. Let us say that a feature map in the shape of $h \times w \times n$ is generated from the previous convolution operations, where n is the number of filters and h and w represent height and width of the feature maps produced, respectively. In this sense, a maximum element can be extracted from each filter so that an $1 \times 1 \times n$ vector is generated. The feature vector is then expanded to emulate the same size of the original feature maps, and then merged with the original feature maps to be convolutionally mapped together.

Our intention of such structure concentrates on reusing the relationship among current positional features and broadcasting them for global comparisons during further convolution operations. Concatenating extracted features with the original feature maps, further convolution filters are able to correlate the objective visuospatial features (convolved features with CCE) and the relative visuospatial features (broadcasting features).

The whole structure of BCN can be succinctly described in an equation as:

$$\mathcal{BCN}(F) = [\mathcal{E}(\max([F, C] * k), h, w), C], \tag{1}$$

where $[\cdot, \cdot]$ refers to the concatenation of feature maps. Here, $F \in \mathbb{R}^{h \times w \times n'}$ is the input feature map for the broadcast convolution module, $C \in \mathbb{R}^{h \times w \times n_c}$ is n_c coordinate planes, $*k$ represents a few layers of convolutional operations whose structure is defined in k (*e.g.* two layers of 1×1 convolution with ReLU activation). Assuming that n is the number of filters in the last layer of successive convolution operations $*k$, the max-pooling operation is taken for each of n output feature maps such that it results in an $1 \times 1 \times n$ vector. $\mathcal{E}$ denotes an

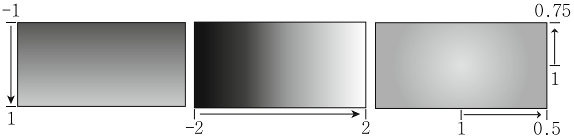

Fig. 2. An example of coordinate planes, c_x, c_y and c_r, for feature maps with 2:1 aspect ratio of width to height. The brighter, the higher value.

expansion operation which copies its input vector to the entire $h \times w$ positions. The proposed BCN outputs a broadcast feature map $B \in \mathbb{R}^{h \times w \times (n+n_c)}$ which is concatenated with the original feature maps F and fed to the next layer as shown in Fig. 1.

The convolution layer(s) right after the CCE phase can be one or multiple, however the major purpose of its(their) presence is to generate abstract representations based on locations. This allows the further convolutional mappings in the later process to infer relative positions among features with the implanted information on the objective positions. In this paper, we have implemented two or three layers of 1×1 convolution after CCE for all experiments to make each pixel in feature maps being convolved depth-wise with undissolved coordinate information. Additionally, the number of feature maps has kept fairly small so that the coordinate planes can be adequately reflected to the outputs after convolution operations. Yet, the number of coordinate planes can apparently be adjusted if needed. This simple setting allows a large improvement on efficiency of extracting and utilizing spatial features in CNNs without much additional computation.

Since CNNs have performed generalization by taking advantage of sharing convolutional kernels in all input locations, their consequential structures are difficult to conserve spatial information throughout the layers. One of the intuitive ways to reflect location information to filtered outputs is to embed unique coordinates into the inputs. In our method, the feature location information is implanted in original feature maps as additional channels. This specific decision comes from the motivation to assign objective positional components to each feature during convolutions before the max-pooling phase, and to establish relative location for the comparison against other spatial features when broadcasting. This furthers the productivity of coordinate embedded visual features and allows additional convolutional mappings to reflect the feature positions when generating higher level features.

Three different coordinate planes are defined as in Fig. 2 and used in this paper; one for the x-axis, c_x, another for the y-axis, c_y and the last one for the radial distance from the center, c_r. Similar to the conventional coordinate feature embedding approaches [22,26,29,32], these planes with the normalized coordinates reduce initial learning bias and provide additional feature location information. Since inputs of our module may not necessarily have the shape of a square, elements of each plane are normalized according to the aspect ratio of inputs. As it can be seen in Fig. 2, the Cartesian coordinates c_x and c_y tend to be

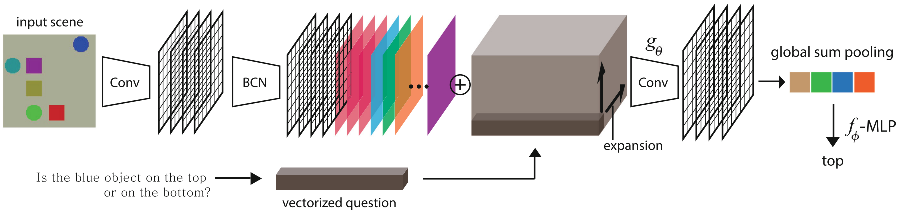

Fig. 3. Multi-Relational Networks. For visual relational reasoning problems, input scenes are fed into ordinary convolutional layers and a BCN module. Since the outputs of BCNs connote visuospatial features of multiple objects, they can be merged with the original features and the expanded question tensor for further convolution operations, g_θ. Then, global sum poolings are operated on feature maps produced, and the outcomes are integrated to answer questions.

zero towards the center. This prevents features from being initially biased near bottom right-side which occurs if coordinates near top left start from zero as used in usual computer graphics. Also, when inputs with different aspect ratios are given, initial values in CCE are scaled to cope with the different aspect ratio.

3 Multi-relational Network

Visuospatial tasks necessitate recognition of relevant features from the spatial organization of stimuli and selection of stimulus that matches one or more of these identified features [3,28]. Algorithms such as relation networks (RNs) [27, 29] have introduced solutions for relational reasoning problems including CLEVR dataset [16] based on their spatial grid features – the dataset is explained in a more detailed manner later in this paper. Utilizing BCN, RNs can be greatly improved in both performance and computational efficiency.

In [29], an RN module for a set of n objects, $O = \{o_1, \cdots, o_n\}$, is introduced, which consists of two functions f_ϕ and g_θ such that:

$$\mathcal{RN}(O) = f_\phi(\sum_i \sum_j g_\theta(o_i, o_j)), \tag{2}$$

where f_ϕ represents multi-layer perceptron (MLP) operations on the visuospatial relation features that have been generated by another MLP operation g_θ from all pairwise combinations of objects. The 'relations' are the outputs of $\mathcal{O}(n^2)$ time of computations from g_θ. Earlier in this paper, we have explained how BCN allows multiple spatial features to be represented in one dimensional vector. With the help of BCN described in (1), original RN in (2) can be revised into Multi-Relational Networks (multiRN) as follows:

$$multi\mathcal{RN}(O) = f_\phi(\sum_i g_\theta(o_i, \mathcal{BCN}(O))). \tag{3}$$

Since $\mathcal{BCN}(O)$ connotes multi-location features for multiple objects, each object feature can be paired with multiple object location features, fed into g_θ and computed in $\mathcal{O}(n)$. This allows not only exponential gain in computation complexity but also relational comparison with multiple objects at once. The outputs of g_θ should, in our context, be redefined as 'multiple relations', and can be trained end-to-end to convey information about how much of certain object feature at particular location should be reflected for visual relational reasoning.

Furthermore, the number of objects, n, in pixel-based relational reasoning problems is defined by $n = h \times w$ where h, w are height and width of resultant 3-dimensional tensors from previous convolution layers. Such trait allows us to replace the MLP operation for g_θ in RNs with 1×1 convolution after channel concatenations of BCN outputs as shown in Fig. 3. The non-pixel-based objects such as questions in visual question and answering (VQA) problems can be additionally concatenated in channels. The final form of our multi-relational network is:

$$multiRN(O) = f_\phi(\sum_i g_\theta(o_i, \mathcal{BCN}(O), question). \tag{4}$$

In practice, *questions* are expanded for channel embedding, and the final tensor is fed into g_θ-convolution operations to achieve 'multiple relations'.

4 Related Works

Our motif has two aspects: 1. extracting and condensing dynamic spatial features and 2. globally projecting them to previous feature maps which largely widen the receptive fields just as demanded in multi-scale reasonings and full resolution dense relational predictions.

Many previous computer vision related researches have used pooling techniques for extracting meaningful spatial features [4]. Spatial pyramid pooling [8] uses manual control over pooling scales and regional features are extracted from variations of hand-engineered pooling scopes. This issue has been developed in [15] where a large set of various pooling bins are initiated and the algorithm learns to select sparse subset of them. These works take hand-crafted pooling regions that cannot be learned end-to-end.

ROI pooling structure [4,5] is similar to our method's in a way of executing pooling methods in channel-wise. Each channel represents feature maps from starting stride point to last, and thus may possess objective spatial features. However, with conventional ROI pooling methods, each feature map does not reflect any relations among feature maps.

While effective receptive field size is previously known to increase linearly with the number of convolutional layers [25], the works in [5] have found that it actually increases with square root of the number of layers which is a much slower rate. This finding further leads to a logical doubt that deep CNNs may not have large enough receptive field size even at top layers. This phenomenon is prominent in fully convolutional networks (FCNs) with a large input image. To overcome the issue, large enough receptive field size is not the only aspect that is essential but the learning of its flexibility depending upon situations, which explains why atrous convolution methods are used widely.

Deformable convolutional neural networks (Deformable ConvNet) [5] is engaged with deformable convolutions that learn the applicable filters with adaptive receptive fields. Their convolution filters are chained with offset parameters which represent the mapping of the original receptive fields to unique and irregularly dispersed receptive fields from each spatial location. The offset parameters are optimized along with the convolutional filters during back-propagation. However, Deformable ConvNets require extra computation upon original CNNs, and their feature map sampling method in a local manner hinders themselves from having complete spatial support. They, as their primitive motivation, further emphasize the importance of adaptive receptive learning in the needs of effectively computing large enough receptive fields at top layers.

Other atrous convolution usages include increasing the receptive field size by sampling from dilated sparse regions [11]. This allows to retain the same computational complexity as previous CNNs' while increasing receptive field sizes, and thus is widely used in semantic segmentation problems [1,24,33]. If depth conditions are excluded, the method is still doubted on utilizing enough size of receptive fields. Our model handles extracted convolutional features by projecting them to global receptive fields, intending to convolve the features from various spatial locations with the original features.

A dominant approach used in the visual relational reasoning domain is the Relation Network module (RN) [29]. As mentioned in Sect. 3, RN learns the relations among objectified features of CNN through pairwise combinations, computations of which increase quadratically with the number of the objects while multiRN gains a comparable improvement on computation efficiency through manipulation of 'multiple relations' induced by the BCN module. A FiLM module [26], as another method for relational reasoning problems, conditions features before activation functions in a similar way as done in LSTM gates [10] and SENET's feature excitations [12]. Such attempt of learning visual feature conditionings for reasoning largely differs from our method of extracting and broadcasting key object features to the entire receptive fields. Also, compared to their suggested model that incorporates multiple residual blocks and large GRUs and MLPs, a multiRN model requires much smaller networks to achieve similar performances.

5 Experiments

The proposed method is tested in several experiments and compared with other similar methods to verify the effectiveness of BCN and MultiRN. The purposes of the experiments are to investigate the followings:

(1) the capability of BCN in representation power of features, (2) the effectiveness of MultiRN on visual reasoning problems, (3) feasible extensions of receptive fields caused by BCN and practical expressions of multiple objects in

multiRN as specified in Sect. 3, and (4) variations of coordinate embeddings in visual features and their performances.

5.1 Tasks

We have experimented our methods on four datasets, Scaled-MNIST, STL-10, Sort-of-CLEVR with pixels, and CLEVR with pixels, for different purposes.

Scaled-MNIST dataset is our own remodeled version of original MNIST dataset [20] to verify that the BCN can effectively handle positional information while globally extending the receptive field. Locating the original MNIST of size 28×28 to 128×128 image space, we randomly scale the width irregularly ranging from 28 to 105 pixels and the aspect ratio from 1:0.8 to 1:1.2. All digits are positioned randomly within the new image space, preserving its complete form. Along with the class labels, another label is added for the locations of center points for each digit in order to evaluate localization performances of the models.

STL-10 [2] is used to evaluate the capability of BCN for natural image classification problems. It is a set of natural image with ten class labels each of which has 500 training images and 800 test images. There are also unlabeled images within the dataset, but we do not use them because experiments on them are irrelevant to our intentions. The data consists of images sized in 96×96 with higher resolution than those of CIFAR-10 but less number of training images.

CLEVR with pixels [16] is one of the visual QA datasets which is a challenging problem set requiring high-level relational reasonings. The dataset contains images of 3-D rendered objects and corresponding questions asking about several attributes of the objects. We have experimented only on the pixel version of the CLEVR dataset whose images are represented in 2-D pixel-wise.

Sort-of-CLEVR with pixels [29] is our main experiment for multiRN. Sort-of-CLEVR is a more simplified version of CLEVR, which is a set of images combined with caption dataset for relation and non-relation reasonings. Each image contains six of differently colored 2-D geometric shapes, and corresponding 20 questions; 10 for relational and 10 for non-relational reasonings. Questions in this dataset are already vectorized, and thus the experiments are independent from any additional vector embedding models, which allows more reasonable comparison based on the results.

5.2 Evaluation of BCN

For all experiments, BCN uses multiple convolution layers with different size of 1×1 kernels and ReLUs for non-linearity. The number of filters in m convolution layers for a BCN is written in the form of $[S_1, \cdots, S_m]$ in each experiment.

Scaled-MNIST: As a baseline model, we have stacked three to five of convolutional layers with 24 of 3×3 kernels, stride size of 2 and padding size of 1 for

Table 1. The model with BCN has the highest classification accuracy and the lowest localization error on the Scaled-MNIST data. Skip-connection denotes using the shape of 1 × 1 convolution layers for BCN. This can be viewed just the same as skip-connection [30]

Model	Classification Acc.	Localization Err.	#Params	Runtime[1]
Baseline	84.4%	0.151	11.2 K	4.5 ms
Baseline(depth 4)	94.9%	0.149	16.5 K	5.1 ms
Baseline(depth 5)	96.4%	0.089	21.9 K	5.5 ms
Baseline(depth 5, 2× filters)	97.2%	0.077	85.2 K	8.5 ms
Base + Deformable Conv [5]	90.8%	0.087	32.0 K	16.7 ms
Base + Dilated Conv [33]	91.0%	0.152	11.2 K	4.7 ms
Base + CCE	84.8%	0.088	11.9 K	4.6 ms
Base + Skip-connection	87.0%	0.151	29.0 K	8.4 ms
Base + CCE + Skip-connection	87.9%	0.071	29.2 K	8.41 ms
Base + BCN w/ average pooling	92.7%	0.064	29.2 K	8.42 ms
Base + BCN	**97.5%**	**0.023**	29.2 K	8.42 ms

each edge, followed by a ReLU activation function and a batch normalization, depending on the comparison target. With the same baseline model (a 3-layered CNN), experiments of deformable convolutions and dilated convolutions are also done. The deformable convolution filters are applied to all three layers, and the dilated convolution filters are applied in the second layer of the baseline model with 2 × 2 dilation. A BCN with an output channel length of $[64, 64, 128]$ is applied once in between the second and third layers. After applying the BCN, the number of feature maps increases, so in order to match the input dimension of the third layer, 1 × 1 convolutions are operated for dimension reduction to 24. This setting is purposefully designed to compare performance of a simpler model with BCN against that of baseline model with more depths and kinds of convolution on the given data. Also, the network structure of expanding channels from 24 to 128 is intended for having more channels when globally max-pooling spatial features to preserve enough visual context information for broadcasting.

Table 1 shows[1] the performance enhancement of our method on both classification and localization results on the Scaled-MNIST dataset. Considering the baseline model achieves 97.8% classification accuracy in our experiment on the original MNIST, its performance of the Scaled-MNIST clearly implies its structurally inherent limitation as the size of the required receptive field increases while making the model deeper to extend the receptive field yields a better result as shown in the table. Even if we increase the receptive field by deepening the convolution net with up to five layers, it can be seen that our method with a shallower depth of three layers reaches a higher performance. The model with BCN even shows better performance than the 5-layer baseline model with twice

[1] All runtimes are the training time measured on Nvidia Titan X (Pascal) GPU and 8 core CPU(i7-6700K) per 100 samples.

Table 2. Results on STL-10. All models are trained from scratch without any external dataset

Model	Accuracy	#Params	Runtime[1]
Tho17-2 Single [31]	75.76%	1.46 M	–
Tho17-2 Ensemble [31]	78.66%	1.46 M	–
Baseline	68.75%	114 K	18.1 ms
Baseline + Skip-connection	69.79%	192.7 K	18.2 ms
Baseline + BCN	**72.18%**	193.2 K	18.2 ms
Resnet18	76.27%	11.2M	102.2 ms
Resnet18 + BCN	**77.00%**	11.46 M	105.5 ms

the number of filters by 0.3% for classification and three times less error for localization.

Furthermore, the significance of the broadcasting phase can be well analyzed from the results. While embedding additional coordinate information reduces the localization error and the skip-connection alone improves the classification performance, compared to the model using both, the model with BCN improves the classification accuracy by 9.6% and reduces the localization error by 0.48. Besides from aspects of CCE and the skip-connection technique, concatenation of expanded max-pooled features of BCN allows the network to globally broaden receptive fields and directly learn visually relational features. And the result of the BCN using average-pooling instead of max-pooling shows a significant performance decline, which indicates that average-pooling is not a suitable method considering that the purpose of max-pooling is to extract key features.

The goal of deformable convolutions is similar to ours in many ways, and the implementation has also led to desirable results, getting better scores on both classification and localization. The model with dilated convolutions has also resulted in an impressive improvement on classification accuracy while conserving the same parameter numbers and computation speed as those of the baseline. Nonetheless, the performance of BCN model outperforms both models. In addition, BCN shows relatively high robustness against deformations compared to other methods, and the experimental results are included in the supplementary material.

STL-10: For a baseline model for the STL-10 dataset, four layers of convolutions with 64 of 3 × 3 kernels with the stride size of 2 and the padding size of 1 for each edge are applied. A ReLU activation function and a batch normalizing operation follow after each convolution layer. A BCN module of size [128, 128, 256] is applied in between the third and the fourth convolution layers. After applying the BCN module, the number of feature maps increases, so in order to match the input dimension of the fourth layer, an 1 × 1 convolution layer is used for dimension reduction to 64. We, in addition, have applied the same size of BCN to a Resnet18 model [9]. The BCN module within the Resnet18 model is applied between the third and the fourth residual blocks. Then, 256 of 1×1 convolution filters are used to reduce number of feature maps, and the entire model is trained end-to-end from scratch.

We compare our model against the baseline model in Table 2 to evaluate the performance of BCN for a natural image classification problem. Because BCN extends the receptive field size to whole image grids, it allows 3.4% of performance enhancement. Also, this is an even better result than the model using the same convolution layers of BCN as a Skip-connection. The additional experiment where BCN is applied within the Resnet18 model shows that our approach sets a new state-of-the-art performance in STL-10 experiment on a single network basis with 77.0% accuracy, which is 1.24% higher than the single model of Tho17-2 [31].

Table 3. Results on Sort-of-CLEVR. RN* is the reproduced result with same model of [29] on a single GPU. RN† is a model of which the network structure is set the same as multiRN except that pairwise comparison for a fair comparison. CNN$_h$ denotes the CNN that has 1 stride for fourth convolution layer instead of 2 stride to handle more objects

Model	Relational	Non-relational	#Params	Runtime[1]
CNN+RN [29]	94%↑	94%↑	19.5 M	-
CNN+RN*	91.0%	99.6%	19.5M	575.8 ms
CNN+RN†	89.9%	99.8%	365K	23.5 ms
CNN$_h$+RN†	96.5%	99.9%	365K	315.6 ms
CNN+MLP	74.2%	65.0%	239 K	6.2 ms
CNN+CCE+MLP	72.9%	64.5%	258 K	6.2 ms
CNN+multiRN w/o BCN	88.7%	99.3%	224 K	7.5 ms
CNN+multiRN	92.9%	99.9%	345 K	8.3 ms
CNN$_h$+multiRN	**96.7%**	**99.9%**	345 K	9.9 ms

5.3 Pixel-Based Relational Reasoning Problem

Model: For relational reasoning experiments, we construct our multiRN model based on the RN model that is used for the CLEVR dataset out of two models reported in [29] which consists of one trained for the Sort-of-CLEVR dataset and a shallower model for the CLEVR. Four convolution layers with 24 of 3×3 kernels, followed by ReLU activations and batch normalizations, are used for the CNN part. MultiRN consists of a BCN of size [128, 128, 256], two convolution layers with 256 1×1 kernels for g_θ, and two MLP layers of 256 units for f_ϕ. Also, to verify that multiRNs use computation resource efficiently, we change the fourth convolution layer's stride from 2 to 1 in the CNN$_h$, which quadruples the number of objects that the following network has to handle. MultiRN, for the CLEVR task, uses LSTM of 128 hidden unit of 2 layers for natural language question processing.

Sort-of-CLEVR: Results of Table 3 suggest that using our version of RN (multiRN) is better than the RN with similar structure at reasoning relations by 3%, which is the main job of Sort-of-CLEVR task, despite of comparably less computational cost. Compared to both RN and multiRN, CNN+MLP model performs far poorly. Even embedding coordinate information into the same model worsens the performance, supposedly, due to overfitting. This implies that the performance enhancement of RN and multiRN is not simply caused by the addition of coordinate maps. For a further ablation study, the BCN module is removed along with the coordinate encoding function, while having g_θ and f_ϕ remained in order to generate the 'CNN+multiRN w/o BCN' model. This model lacks information of correlations among multiple objects, and thus performs 4.2% lower than the CNN+multiRN model. To check the efficiency when the number of object inputs upsurges for the multiRN to manage, we reduce the stride of the last layer of the preceding CNN from 2 to 1, denoting as CNN$_h$. Since the computation complexity of multiRN is $\mathcal{O}(n)$, which is a strong advantage over RNs with $\mathcal{O}(n^2)$ for n number of objects, the multiRN's computation cost is only quadrupled (25 to 100) for CNN$_h$ while RN takes 16 times greater computation loads (625 to 10000). Both RN and multiRN with CNN$_h$ have achieved impressive performance gain with increase in the number of feature objects, but there is a large difference in computational efficiency.

Table 4. Results on CLEVR from Pixel. RN* is the result when we reproduce the same model as the paper [29] on a single GPU. ‡ denotes the result of changing the size of an input image from 128 × 128 to 224 × 224, which is the same as FiLM [26]

Model	Overall	Count	Exist	Compare numbers	Query attribute	Compare attribute
Human [17]	92.6	86.7	96.6	86.5	95.0	96.0
Q-type baseline [17]	41.8	34.6	50.2	51.0	36.0	51.3
LSTM [17]	46.8	41.7	61.1	69.8	36.8	51.8
CNN+LSTM [17]	52.3	43.7	65.2	67.1	49.3	53.0
CNN+LSTM+SA [17]	68.5	52.2	71.1	73.5	85.3	52.3
CNN+LSTM+RN [29]	95.5	90.1	97.8	93.6	97.9	97.1
CNN+LSTM+RN*	90.9	86.7	97.4	90.0	90.2	93.5
CNN+GRU+FiLM with ResNet-101 [26]	**97.7**	94.3	99.1	96.8	99.1	99.1
CNN+GRU+FiLM from raw pixels [26]	97.6	94.3	**99.3**	93.4	**99.3**	**99.3**
CNN+LSTM+ multiRN	92.3	85.2	96.5	93.6	95.1	92.9
CNN$_h$+LSTM+ multiRN	97.2	94.1	98.9	**98.3**	98.6	97.6
CNN$_h$+LSTM+ multiRN‡	**97.7**	**94.9**	99.2	97.2	98.7	98.3

CLEVR: An experiment has been done on a more challenging relational reasoning problem to test the performance of multiRN compared to existing methods.

The model is replicated having internal RN module replaced with multiRN to create ‘CNN+LSTM+MultiRN’. The structure of multiRN model is the same as it was used for Sort-of-CLEVR dataset, and question vectors are generated from an LSTM model which is also implemented in the original RN module.

The results of the comparative experiments on CLEVR are shown in Table 4. The RN*, our reproduced version of the relational network from [29], has not been able to achieve the same performance as in the paper due to lack of original version’s details of hyper-parameters and additional network control factors. However, since the input convolutional feature maps and question vectors are generated with the same CNN+LSTM settings, reasonable comparisons can still be made among the candidate methods. For this problem, our implementation using multiRN performs 1.4% better than RN*, and multiRN with CNN$_h$ achieves even far better performance enhancement of 6.3%. This is an impressive result considering that the same CNN+LSTM+RN model architecture is used as in RN* and allows 95.5% of performance whilst both of multiRN and multiRN with CNN$_h$ require a much smaller amount of computation. We have not been able to proceed an experiment with the CNN$_h$+LSTM+RN model because of the out-of-memory problem when dealing with an increased number of objects. Furthermore, our model achieves 97.7% of test accuracy by changing the size of an input image to 224 × 224 which is the same as it is in FiLM [26]. For the best of our knowledge, this score is the state-of-the-art performance on CLEVR with raw pixels, and is compatible with FiLM using ResNet-101 [9] pre-trained on ImageNet [6].

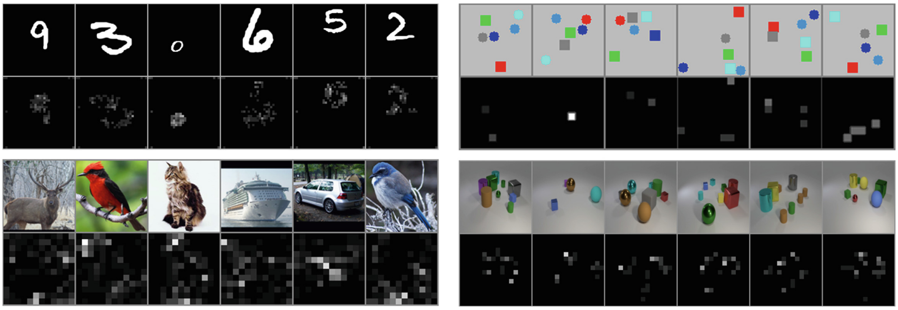

Fig. 4. Activation map shows how much information is broadcasted at each location. From top to bottom are Scaled-MNIST and STL-10 on the left, Sort-of-CLEVR and CLEVR on the right. More activation maps are included in the supplementary material.

5.4 Can BCN Extend the Receptive Field Globally and Represent Multiple Objects?

For better understanding of our model, visualization of activation maps for different images of each experiment is provided in Fig. 4. Activation maps are acquired by unsampling the global max-pooling layer, and simply masking the pixel as 1

Table 5. Results on coordinate embedding methods. Our method outperforms conventional coordinate methods in both Scaled-MNIST and Sort-of-CLEVR

	Scaled-MNIST		Sort-of-CLEVR	
Method	Accuracy	Localization err.	Relational acc.	Non-relational acc.
No Coordinate channels	92.5%	0.151	91.4%	91.8%
X, Y Coordinates with top-left zero	96.9%	0.029	92.6%	99.9%
X, Y Coordinates with zero-centered	97.5%	0.025	96.1%	99.9%
X, Y Coordinates with zero-centered + Radial distance	97.5%	0.023	96.7%	99.9%

from where it was chosen. The number of masks is the same as the number of 1×1 convolution filters, and summing them through feature dimension outputs of the activation map. In activation maps for Scaled-MNIST, we can intuitively find that the activation map corresponds well with each digit's location in the original image. The Scaled-MNIST dataset may have too many zero inputs for a network to easily respond to spatial features, but activation maps in STL-10 apparently show that BCN makes abstractions of features well even in natural images. Notably, we could observe that activation maps tend to draw features from important locations, such as edges, faces, legs and so on. As this information is broadcast to all locations, the receptive field expands globally. For Sort-of-CLEVR task and CLEVR task, we can obviously see that BCN is well trained to make abstractions of multiple objects in their image. This means that the 'multiple relations' of multiRN described in Sect. 3 is established through BCN. Note that since activation maps only represent the maxpooled features, it does not have to include all objects by themselves. Output convolution features that are inputted to the BCN module, which will be further combined with BCN output, also contain features of objects.

5.5 Study on Coordinate Embedding Methods

Our demonstration on the effects of coordinate channels is shown in Table 5. The result without coordinate channels has far less score compared to the result with the coordinate channels embedded. The model of coordinates with zero-centered shows a significantly higher relational accuracy than the model of conventional coordinates with top-left zero for Sort-of-CLEVR, but it is slightly worse than that of three planes with extra radial distance plane included. Figure 5 shows the absolute values of the output from the convolution kernel with randomly initialized weights passing through the coordinate channels. As shown, a coordinate system with a conventional top-left of zero is deflected diagonally, but a coordinate system with zero center is deflected to the outside. On the other hand, by adding coordinate channels with radial distances, it can be seen that most of the deflection can be removed while providing additional coordinate information. Furthermore, although we have expected better feature representation and localization performance by concatenating an additional coordinate plane and reducing initial weight bias, the performance gain has only occurred in the localization aspect. This evidently implies that convolution operations are inherently biased towards the center with Gaussian distribution [25], and thus the additional coordinate plane have not been a critical catalyst.

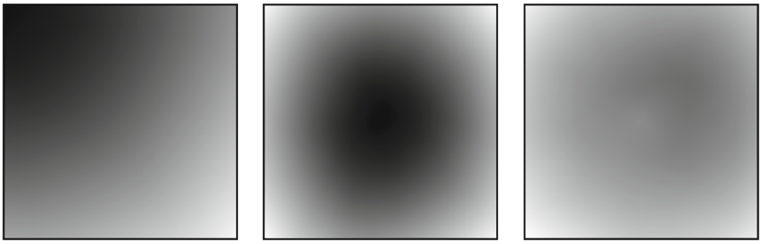

Fig. 5. The outputs of coordinate planes through the convolution kernel. Left: X, Y coordinates with top-left zero. Center: X, Y coordinates with zero-centered. Right: X, Y coordinates with zero-centered and additional Radial channel. The right has the least deflection.

6 Conclusion

We have shown that utilizing Broadcasting Convolutional Network (BCN) allows conventional CNNs to effectively collect and represent spatial information with efficient extension of receptive fields, which results in remarkable performance on localization problems. With BCN's ability of representing compounded spatial features in all receptive fields, we have proposed Multi-Relational Networks that greatly improve RN [29] in terms of computational gains while achieving a state-of-the-art performance in pixel-based relation reasoning problems.

In future works, we intend to study whether BCN can be applied to other domains such as object detection or semantic segmentation. And we need to study applying multiRN to various problems that require visual relational reasoning.

Acknowledgement. This work was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (2017M3C4A7077582).

References

1. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. arXiv preprint arXiv:1606.00915 (2016)

2. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 215–223 (2011)
3. Crone, E.A., Wendelken, C., Van Leijenhorst, L., Honomichl, R.D., Christoff, K., Bunge, S.A.: Neurocognitive development of relational reasoning. Dev. Sci. **12**(1), 55–66 (2009)
4. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems, pp. 379–387 (2016)
5. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. arXiv preprint arXiv:1703.06211 (2017)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR 2009 (2009)
7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
8. He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, Sun, Jian: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Fleet, David, Pajdla, Tomas, Schiele, Bernt, Tuytelaars, Tinne (eds.) ECCV 2014. LNCS, vol. 8691, pp. 346–361. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10578-9_23
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
11. Holschneider, M., Kronland-Martinet, R., Morlet, J., Tchamitchian, P.: A real-time algorithm for signal analysis with the help of the wavelet transform. In: Wavelets, pp. 286–297. Springer, Heidelberg (1990)
12. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. arXiv preprint arXiv:1709.01507 (2017)
13. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in Neural Information Processing Systems, pp. 2017–2025 (2015)
14. Jeon, Y., Kim, J.: Active convolution: learning the shape of convolution for image classification. arXiv preprint arXiv:1703.09076 (2017)
15. Jia, Y., Huang, C., Darrell, T.: Beyond spatial pyramids: receptive field learning for pooled image features. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3370–3377. IEEE (2012)
16. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C.L., Girshick, R.: Clevr: a diagnostic dataset for compositional language and elementary visual reasoning. arXiv preprint arXiv:1612.06890 (2016)
17. Johnson, J., et al.: Inferring and executing programs for visual reasoning. arXiv preprint arXiv:1705.03633 (2017)
18. Kemp, C., Tenenbaum, J.B.: The discovery of structural form. Proc. Natl. Acad. Sci. **105**(31), 10687–10692 (2008)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
20. LeCun, Y.: The MNIST database of handwritten digits (1998). http://yann.lecun.com/exdb/mnist/
21. LeCun, Y., Bengio, Y.: Convolutional networks for images, speech, and time series. Handb. Brain Theory Neural Netw. **3361**(10), 1995 (1995)

22. Liang, X., Wei, Y., Shen, X., Yang, J., Lin, L., Yan, S.: Proposal-free network for instance-level object segmentation. arXiv preprint arXiv:1509.02636 (2015)
23. Logan, G.D., Sadler, D.D.: A computational analysis of the apprehension of spatial relations (1996)
24. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
25. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 4898–4906 (2016)
26. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: visual reasoning with a general conditioning layer. arXiv preprint arXiv:1709.07871 (2017)
27. Raposo, D., Santoro, A., Barrett, D., Pascanu, R., Lillicrap, T., Battaglia, P.: Discovering objects and their relations from entangled scene representations. arXiv preprint arXiv:1702.05068 (2017)
28. Raven, J.C.: Standardization of progressive matrices, 1938. Psychol. Psychother. Theory, Res. Pract. **19**(1), 137–150 (1941)
29. Santoro, A., et al.: A simple neural network module for relational reasoning. arXiv preprint arXiv:1706.01427 (2017)
30. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks. In: Advances in Neural Information Processing Systems, pp. 2377–2385 (2015)
31. Thoma, M.: Analysis and optimization of convolutional neural network architectures. arXiv preprint arXiv:1707.09725 (2017)
32. Tian, Y., Zhu, Y.: Better computer go player with neural network and long-term prediction. arXiv preprint arXiv:1511.06410 (2015)
33. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)